

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-025042

(43)Date of publication of application : 29.01.1999

(51)Int.Cl. G06F 15/00
G06F 15/00
G06F 13/00
G06F 13/00

(21)Application number : 10-123131

(71)Applicant : INTERNATL BUSINESS MACH CORP <IBM>

(22)Date of filing : 06.05.1998

(72)Inventor : CUOMO GENARO A
NGUYEN BINH Q
NGUYEN HUNG D
SINGHAL SANDEEP K

(30)Priority

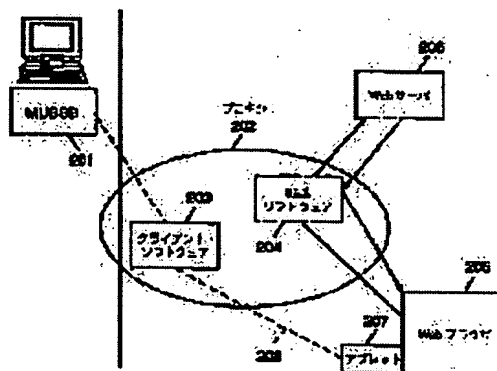
Priority number : 97 855244 ...Priority date : 13.05.1997 ...Priority country : US

(54) METHOD AND SYSTEM FOR REALIZING COGNITION, TOUR AND CONFERENCE IN WWW IN PORTABLE FORM BY USING PROXY AND SHARED STATE SERVER

(57)Abstract:

PROBLEM TO BE SOLVED: To realize collaboration in different execution environments in a portable form by operating collaboration in an existing Web browser by using a proxy and a shared state server.

SOLUTION: A proxy 202 communicating with an MUSSS(multi-user shared state server) 201 receives a Web page request from a local Web browser 205, and transfers it to a Web server 206 at a corresponding designation. A web browser 205 displays data obtained from the Web server 206 through transfer software 204. The browser 205 executes an uplet 207 integrated into data obtained by the transfer software 204. The uplet 207 maintains connection 208 with a client software 203 of the proxy 202. The uplet 207 receives a command from a cooperative session through the connection 208, and applies information which the client software 203 can distribute to the cooperative session.



LEGAL STATUS

[Date of request for examination]

23.10.1998

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平11-25042

(43) 公開日 平成11年(1999) 1月29日

(51) Int.Cl. ⁶	識別記号	F I
G 0 6 F 15/00	3 1 0	G 0 6 F 15/00
	3 9 0	
13/00	3 5 5	13/00
	3 5 7	

審査請求 有 請求項の数33 OL (全 13 頁)

(21) 出願番号 特願平10-123131

(22) 出願日 平成10年(1998) 5月6日

(31) 優先権主張番号 08/855244

(32) 優先日 1997年5月13日

(33) 優先権主張国 米国 (US)

(71) 出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MACHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(72) 発明者 ジェナロ・エイ・カオモ

アメリカ合衆国27502、ノース・カロライ
ナ州アベックス、ディア・パレー・ドリベ
ット 112

(74) 代理人 弁理士 坂口 博 (外1名)

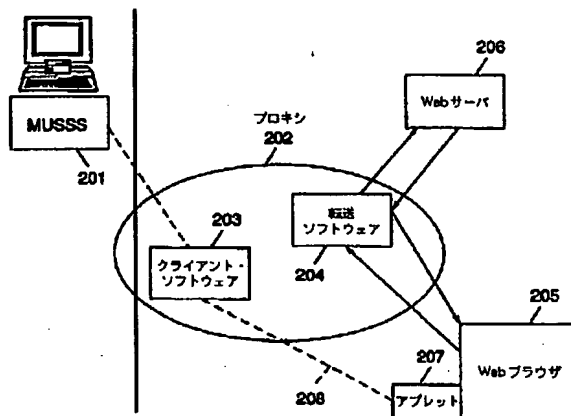
最終頁に続く

(54) 【発明の名称】 プロキシ及び共有状態サーバを使用してWWWでの認知、ツアー及び会議をポータブルに可能にする方法及びシステム

(57) 【要約】

【課題】 WWWコンテンツに共同 (コラボレーショ
ン) 機能 (情報の "プッシュ" と "プル" の両方) がポータ
ブルに追加される。

【解決手段】 特に "プル"・メカニズムを介して任意の
Webコンテンツにアクセスする自由を保つ "プッシュ"
・メカニズムを介してデータを取得するために、共有状
態サーバとの一貫した接続が維持される。更にコラボレ
ーションはポータブルな形で、他の全てのWebコンテ
ンツがハードウェア・プラットフォーム全体及びブラウ
ザの実現形態全体でポータブルであるのと同じ形で実現
される。



【特許請求の範囲】

【請求項 1】 情報サーバからデータを取得する方法であって、

リクエストが情報サーバからのデータのリクエストを開始するステップと、

前記リクエストをプロキシに送るステップと、

前記プロキシが前記リクエストを前記情報サーバに転送するステップと、

前記プロキシが前記情報サーバからデータを受信し格納するステップと、

前記データにアプレットを挿入するステップと、

前記データと前記アプレットを前記データの前記リクエストに配信するステップと、

前記アプレットを実行するステップと、

を含む、方法。

【請求項 2】 前記アプレットを実行する前記ステップは、

クライアント・ソフトウェアとの接続を開始するステップと、

クライアント・ソフトウェアとの前記接続を終了するステップと、

を含む、請求項 1 記載の方法。

【請求項 3】 addURL 通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、

請求項 2 記載の方法。

【請求項 4】 removeURL 通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、

請求項 2 記載の方法。

【請求項 5】 readData 通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、

請求項 2 記載の方法。

【請求項 6】 writeData 通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、

請求項 2 記載の方法。

【請求項 7】 changePage 通知をクライアント・ソフトウェアとの前記接続を通して受信するステップと、

前記 changePage 通知に従って新しいデータの取得を開始するステップと、を含む、請求項 2 記載の方法。

【請求項 8】 前記クライアント・ソフトウェアからターゲット・リーダのフォロー・リクエストを開始するステップと、

前記フォロー・リクエストの発信側の ID を記録するステップと、

前記ターゲット・リーダによって行われる操作の結果として前記発信側への changePage リクエストを開始するステップと、

を含む、請求項 7 記載の方法。

【請求項 9】 前記クライアント・ソフトウェアはマルチユーザ共有状態サーバ (MUSSS) と通信する、請求項 2 記載の方法。

【請求項 10】 クライアント・ソフトウェアとの接続を開始する前記ステップは、グローバル・アプレットの ID を確認するステップを含み、前記クライアント・ソフトウェアとの前記接続は、前記グローバル・アプレットにリクエストを転送することによって確立される論理接続である、請求項 2 記載の方法。

【請求項 11】 グローバル・アプレットの有無を監視するステップと、

前記グローバル・アプレットが存在しない場合は前記グローバル・アプレットになるステップと、

を含む、請求項 10 記載の方法。

【請求項 12】 情報サーバからデータを取得するシステムであって、

リクエストが情報サーバからのデータのリクエストを開始するための手段と、

前記リクエストをプロキシに送る手段と、

前記プロキシが前記リクエストを前記情報サーバに転送するための手段と、

前記プロキシが前記情報サーバからデータを受信し格納するための手段と、

前記データにアプレットを挿入する手段と、

前記データと前記アプレットを前記データの前記リクエストに配信する手段と、

前記アプレットを実行する手段と、

を含む、方法。

【請求項 13】 前記アプレットを実行する前記手段は、クライアント・ソフトウェアとの接続を開始する手段と、

クライアント・ソフトウェアとの前記接続を終了する手段と、

を含む、請求項 12 記載のシステム。

【請求項 14】 クライアント・ソフトウェアとの前記接続を通して addURL 通知を送る手段を含む、

請求項 13 記載のシステム。

【請求項 15】 クライアント・ソフトウェアとの前記接続を通して removeURL 通知を送る手段を含む、

請求項 13 記載のシステム。

【請求項 16】 クライアント・ソフトウェアとの前記接続を通して readData リクエストを送る手段を含む、

請求項 13 記載のシステム。

【請求項 17】 クライアント・ソフトウェアとの前記接続を通して writeData リクエストを送る手段を含む、

請求項 13 記載のシステム。

【請求項 18】 クライアント・ソフトウェアとの前記接続を通して changePage 通知を受信する手段と、

前記 changePage 通知に従って新しいデータの取得を開始する手段と、

を含む、請求項 13 記載のシステム。

【請求項 19】 前記クライアント・ソフトウェアからターゲット・リーダのフォロー・リクエストを開始する手

段と、
前記フォロー・リクエストの発信側の ID を記録する手段と、
前記ターゲット・リーダによって行われる操作の結果として前記発信側への changePage リクエストを開始する手段と、
を含む、請求項 1 8 記載のシステム。

【請求項 2 0】前記クライアント・ソフトウェアはマルチユーザ共有状態サーバと通信する、
請求項 1 3 記載のシステム。

【請求項 2 1】クライアント・ソフトウェアとの接続を開始する前記手段は、グローバル・アプレットの ID を確認する手段を含み、前記クライアント・ソフトウェアとの前記接続は、前記グローバル・アプレットにリクエストを転送することによって確立される論理接続である、請求項 1 3 記載のシステム。

【請求項 2 2】グローバル・アプレットの有無を監視する手段と、
前記グローバル・アプレットが存在しない場合は前記グローバル・アプレットになる手段と、
を含む、請求項 2 1 記載のシステム。

【請求項 2 3】情報サーバからデータを取得するためにコンピュータ可読媒体に記録されるコンピュータ・プログラム・プロダクトであって、
リクエストが情報サーバからのデータのリクエストを開始するためのコンピュータ可読手段と、
前記リクエストをプロキシに送るコンピュータ可読手段と、
前記プロキシが前記リクエストを前記情報サーバに転送するためのコンピュータ可読手段と、
前記プロキシが前記情報サーバからデータを受信し格納するためのコンピュータ可読手段と、
前記データにアプレットを挿入するコンピュータ可読手段と、
前記データ及び前記アプレットを前記データの前記リクエストに配信するコンピュータ可読手段と、
前記アプレットを実行するコンピュータ可読手段と、
を含む、コンピュータ・プログラム・プロダクト。

【請求項 2 4】前記アプレットを実行する前記コンピュータ可読手段は、
クライアント・ソフトウェアとの接続を開始するコンピュータ可読手段と、
クライアント・ソフトウェアとの前記接続を終了するコンピュータ可読手段と、
を含む、請求項 2 3 記載のコンピュータ・プログラム・プロダクト。

【請求項 2 5】クライアント・ソフトウェアとの前記接続を通して addURL 通知を送るコンピュータ可読手段を含む、
請求項 2 4 記載のコンピュータ・プログラム・プロダク

ト。

【請求項 2 6】クライアント・ソフトウェアとの前記接続を通して removeURL 通知を送るコンピュータ可読手段を含む、

請求項 2 4 記載のコンピュータ・プログラム・プロダクト。

【請求項 2 7】クライアント・ソフトウェアとの前記接続を通して readData リクエストを送るコンピュータ可読手段を含む、

10 請求項 2 4 記載のコンピュータ・プログラム・プロダクト。

【請求項 2 8】クライアント・ソフトウェアとの前記接続を通して writeData リクエストを送るコンピュータ可読手段を含む、

請求項 2 4 記載のコンピュータ・プログラム・プロダクト。

【請求項 2 9】クライアント・ソフトウェアとの前記接続を通して changePage 通知を受信するコンピュータ可読手段と、

20 前記 changePage 通知に従って新しいデータの取得を開始するコンピュータ可読手段と、

を含む、請求項 2 4 記載のコンピュータ・プログラム・プロダクト。

【請求項 3 0】前記クライアント・ソフトウェアからターゲット・リーダのフォロー・リクエストを開始するコンピュータ可読手段と、
前記フォロー・リクエストの発信側の ID を記録するコンピュータ可読手段と、

30 前記ターゲット・リーダによって行われる操作の結果として前記発信側への changePage リクエストを開始するコンピュータ可読手段と、

を含む、請求項 2 9 記載のコンピュータ・プログラム・プロダクト。

【請求項 3 1】前記クライアント・ソフトウェアはマルチユーザ共有状態サーバと通信する、請求項 2 4 記載のコンピュータ・プログラム・プロダクト。

【請求項 3 2】クライアント・ソフトウェアとの接続を開始する前記コンピュータ可読手段は、グローバル・アプレットの ID を確認するコンピュータ可読手段を含み、前記クライアント・ソフトウェアとの前記接続は、前記グローバル・アプレットにリクエストを転送することによって確立される論理接続である、請求項 2 4 記載のコンピュータ・プログラム・プロダクト。

【請求項 3 3】グローバル・アプレットの有無を監視するコンピュータ可読手段と、
前記グローバル・アプレットが存在しない場合は前記グローバル・アプレットになるコンピュータ可読手段と、
を含む、請求項 3 2 記載のコンピュータ・プログラム・プロダクト。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、一般的にはコンピュータ・ソフトウェアに関し、特にWWW（ワールドワイド・ウェブ）でマルチユーザ・コラボレーション（共同操作）を可能にする方法及びシステム、中でも認知、ツアー及び会議をポータブルな形で可能にする方法及びシステムに関する。

【0002】

【従来の技術】現在、供給者から消費者にデータを配信するために用いられている手法は、“プッシュ”及び“プル”の2つである。PointCast、Castanet、TIB等のサービスによって用いられている“プッシュ”・モデルでは、情報はソースで利用できるようになったときには受信側に送られる。Excite等のサービスによって用いられている“プル”・モデルでは、データは明示的リクエストに回答して受信側に転送されるだけである。これら2つの方法は、ランダム・サーチのサポート（“プル”）から動的更新情報の配信（“プッシュ”）まで、様々なデータ取得タスクで他より優れている。

【0003】しかしこれらデータ取得メタフォは、インターネット上、複数のユーザ間でリアルタイム・コラボレーションをサポートするには不十分である。このようなアプリケーションには、共有ホワイトボード、テキスト及びオーディオ・チャット、共有スライド・ショー、ビデオ、通信教育及びWebページの共同ブラウズ等がある。これらのアプリケーションはユーザ・グループを確立し、情報を交換して、あるグループの全てのメンバーが一貫した情報を見ること（例えば同じホワイトボード・イメージを見る、互いのテキスト／オーディオを見る、スライド／ビデオを同期した形で見る、同じWebページを見る等）ができるようにする必要性にその特徴がある。実際、共有状態の一貫した更新内容が各クライアントに“プッシュ”される。同時に、各ユーザが任意のWebページをダウンロード（“プル”のメタフォ）し、その情報を共同型のアプリケーションとグループに提示することによって情報を取得する自由を与える必要がある。

【0004】通常、こうした共同システムはスタンドアロン型アプリケーションとして実行されるが、これは、具体的なデータ・タイプとそれらの特別な“プッシュ”及び“プル”の要件を交換する必要があるからである。確かにこうしたアプリケーションはそれぞれのカスタム・ユーザ・インタフェースを提供するが、既存のツールとの相互運用はできない。

【0005】アメリカ・オンライン（America Online）のVP（Virtual Places）のようなシステムは、コラボレーションをNetscape Navigator等既存のWWWブラウザに統合する試みを行っている。コラボレーションのためにユーザが新しいWebページまたはスライドを見る必要があるとき、このシステムはWebブラウザと直接

通信し、特定のWebページを表示することをブラウザに指示する。この通信を行うために、VPは特定のWebブラウザによりエクスポートされたDDE（動的データ交換）インタフェースに依存する。アプリケーションはDDEにより、特定のエンドポイント・アプリケーションによってサポートされるように、定義済みイベント・セットを交換することができる。

【0006】DDEを使用することは、しかしVPの機能を制限することである。第1にDDEはMicrosoft Windowsプラットフォームでしか利用できない。更にDDEインタフェースはアプリケーションが限定されるので、VPによって用いられる特定のインタフェースしかNetscape Navigatorブラウザとは共同しない。まとめるとVPシステムは、WWWとのコラボレーションを統合するポータブルなメカニズムは提供しない。

【0007】従ってコラボレーション機能（情報の“プッシュ”及び“プル”の両方）をWWWコンテンツにポータブルに追加する方法及びシステムが必要である。特に“プル”・メカニズムを介して任意のWebコンテンツにアクセスする自由を保つ“プッシュ”・メカニズムを介してデータを取得するために、共有状態サーバとの一貫した接続を維持する必要がある。更に、他の全てのWebコンテキストがハードウェア・プラットフォーム全体及びブラウザの実現形態全体の両方でポータブルであるのと同じ形で、コラボレーションをポータブルな形で実現する必要がある。

【0008】本発明の目的は、コラボレーション環境の外部の任意情報へのアクセスを有効にすると同時に、共有状態システムへの一貫したリンクを維持する方法を提供することである。

【0009】本発明の他の目的は、共同型アプリケーションが既存の（非共同型）アプリケーションと、これら既存のアプリケーションを変更せずに対話できるようにすることである。

【0010】本発明の他の目的は、異なるハードウェア、オペレーティング・システム、及び実行環境でコラボレーションをポータブルな形でできるようにすることである。

【0011】前記の目的を達成するため、またここで概略を述べる本発明の目的に従って、プロキシ及び共有状態サーバを使用して既存のWebブラウザ内でコラボレーションを可能にする方法及びシステムが開示される。

【0012】図1を参照する。本発明を実現するために利用できるデータ処理システム8が示してある。データ処理システム8には、LAN（ローカル・エリア・ネットワーク）10、32等の複数のネットワークを追加できる。ネットワークはそれぞれ、好適には複数の個別コンピュータ12、30を含む。もちろん当業者には明らかのように、ホスト・プロセッサに接続された複数のIWS（高性能ワークステーション）をこのようなネット

ワークそれぞれに利用することもできる。前記ネットワークはそれぞれ、共有メモリ、共有記憶装置、相互接続ネットワーク等の通信媒体を介して接続される複数のプロセッサで構成できる。このようなデータ処理システムでは一般的であるが、独立した各コンピュータは、記憶装置14やプリンタ/処理装置16に接続でき、マウス17等のポインティング・デバイスを備えることもできる。

【0013】データ処理システム8はまた、メインフレーム・コンピュータ18等、好適には通信リンク22によりLAN10に接続できる複数のメインフレーム・コンピュータを追加できる。メインフレーム・コンピュータ18はまた、LAN10のリモート記憶装置としても使用できる記憶装置20に接続できる。同様にLAN10は通信リンク24を介して、サブシステム制御装置/通信コントローラ26及び通信リンク34を通して、ゲートウェイ・サーバ28に接続できる。ゲートウェイ・サーバ28は好適にはLAN32をLAN10にリンクする役割を持つIWSである。

【0014】LAN32及びLAN10に関しては、リソース・オブジェクトのリソース・マネージャまたはライブラリ・サービスが格納されていると、複数の文書またはリソース・オブジェクトを記憶装置20内に格納でき、メインフレーム・コンピュータ18によって制御できる。もちろん当業者には明らかなように、メインフレーム・コンピュータ18はLAN10から地理的に遠く離れた場所に置き、同様にLAN10はLAN32からかなりの距離離れていてもよい。例えばLAN32はカリフォルニアに、LAN10はノース・カロライナに、メインフレーム・コンピュータ18はニューヨークに置ける。

【0015】本発明を採用したソフトウェア・プログラム・コードは通常、開発者が配布のためにアクセスするスタンドアロン型ワークステーションまたはLANサーバの記憶装置14のメモリに格納される。ソフトウェア・プログラム・コードは、ディスクまたはCD-ROM等、データ処理システムと併用されるこれまでの様々な媒体に組み込むことができ、また1つのコンピュータ・システムのメモリから何らかのネットワークを通して他のコンピュータ・システムに、その他のシステムのユーザが使用するように配布することもできる。ソフトウェア・コードを媒体に組み込む、或いはまたソフトウェア・コードを配布するこのような手法及び方法は周知の通りであり、ここでは詳細にはふれない。

【0016】図2を参照する。図のとおりWebブラウザと統合される共同システムの基本要素が示してある。MUSSS（マルチユーザ共有状態サーバ）201は1組のユーザを代表して共同サービスを提供する。このサーバは共同者（コラボレータ）間の情報交換を促進する役割を担う。図示していないが、MUSSSは複数の相

互接続サーバで構成でき、それらは共にMUSSSによって用意されるサービスを提供することは理解されよう。

【0017】プロキシ202は各クライアント・ホスト上で動作する。プロキシ202はMUSSS201と通信するクライアント・ソフトウェア203を含む。プロキシは好適な実施例ではクライアント・ホストに存在するが、他の実施例ではプロキシを周知のサーバ上等他のホストに置くこともできる。この通信でクライアントは共同セッションに情報を提供し、共同する他の参加者から情報を受信する。プロキシ202はまた、ローカルWebブラウザ205からWebページ・リクエストを受け取り、対応する宛先のWebサーバ206に転送する転送ソフトウェア204を含む。転送ソフトウェア204は、取得されたデータをWebブラウザ205に配信する前に操作する。

【0018】最後にWebブラウザ205は、転送ソフトウェア204を介してWebサーバ206から取得されたデータを表示する。ブラウザ205は、転送ソフトウェア204によって取得されたデータに組み込まれたアプレット207を実行する。アプレット207はプロキシ202のクライアント・ソフトウェア203との接続208を維持する。接続208上、アプレットは共同セッションからコマンドを受信し、クライアント・ソフトウェア203が共同セッションに配布できる情報を与える。通常、アプレット207はWebブラウザにソフトウェアを、ブラウザ及びプラットフォームに依存しない形で配信するための標準的なメカニズムであるプログラミング言語、JavaまたはJavaScriptで書かれる。

【0019】図3を参照する。WebブラウザによってWebデータがどのように取得されるかを表すフローチャートが示してある。ブロック301でWebブラウザは、プロキシ内部に組み込まれた転送ソフトウェアにデータ・リクエストを転送する。ブロック302で転送ソフトウェアは対応するWebサーバとの接続を開始する。ブロック303でWebサーバはリクエストを受信し転送ソフトウェアにデータを転送する。ブロック304で転送ソフトウェアはWebサーバからデータを受信し、データにアプレットを挿入する。ブロック305で転送ソフトウェアは取得されたデータを（挿入されたアプレットと共に）ブラウザに渡す。ブロック306でブラウザはデータを受信、解析し、データを表示してアプレットを実行する。この手順は307で終了する。図示していないが、ブロック303、304及び305は、データがWebサーバから転送ソフトウェア、Webブラウザへと流れるので、同時に生じることもあることは理解されよう。アプレットは好適な実施例では転送ソフトウェアによって挿入されるが、他の実施例ではWebサーバによってアプレットが挿入されるか、またはページ内に静的に存在してもよい。

【0020】アプレットは、Webブラウザに任意のコンテンツがロードされていても、Webブラウザがコラボレーションとリンクできるように、クライアント・ソフトウェア（図2参照）との接続を開始する役割を担う。アプレットがWebサーバから取得されたデータに挿入されると、転送ソフトウェアは、アプレットをクライアント・ソフトウェアにどのように接続するかを指定するパラメータを与えることができる。またWebページに同時に表示できる複数の文書を表す"フレーム"が含まれるとき、複数のアプレットを同時にアクティブにすることができる。その結果、最初に実行されるアプレットは自体をグローバル変数に置き、そのため複数のアプレットは1つのWebブラウザ内からクライアント・ソフトウェアと同時に通信することはない。

【0021】アプレットは一度実行されると、"addURL"メッセージを送り、ブラウザ・ウィンドウで現在どのページが表示されているかをクライアント・ソフトウェアに通知する。アプレットは終了すると"removeURL"メッセージを送り、ページが表示されなくなったことをクライアント・ソフトウェアに通知する。一貫した情報を取得しました格納するために、それぞれ"readData"、"writeData"のメッセージをプロキシとの間で送信できる。最後にアプレットは、例えば新しいWebページをブラウザにロードするリクエスト等、クライアント・ソフトウェアからのコマンドをチェックする。

【0022】図4を参照する。Webブラウザで実行されるときに転送ソフトウェアによって挿入されたアプレットによって実行される操作を表すフローチャートが示してある。判断ブロック401でアプレットはグローバル変数がヌルかどうか確認する。判断ブロック401の応答がYesの場合、ブロック402でアプレットはグローバル変数を初期化して自体をポイントする。ブロック403でアプレットは転送ソフトウェアによって与えられた接続パラメータを取得する。ブロック404でアプレットはクライアント・ソフトウェアへの接続を開始する。ブロック405でアプレットは"addURL"メッセージを、現在のページのURLをパラメータとしてその接続を通してクライアント・ソフトウェアに送る。この時点でアプレットはブロック406、408及び410を通して3つの同時実行パスを開始している。図5を参照する。判断ブロック406でアプレットは新しいURLをロードするコマンドがクライアント・ソフトウェアから受信されたかどうか確認する。判断ブロック406への応答がYesならブロック407でアプレットは新しいページをロードすることをブラウザに通知し、次に判断ブロック406に戻る。同時にブロック408（図6）でアプレットは、それが他のアプレットからURLを追加するまたは削除するリクエストを受信したかどうか確認する。判断ブロック408への応答がYesならブロック409でアプレットは"addURL"または"removeU

RL"のメッセージをクライアント・ソフトウェアとのその接続を通して送り、判断ブロック408に戻る。同時に判断ブロック410（図7）でアプレットはブラウザから停止メッセージが受信されたかどうか確認する。判断ブロック410への応答がNoなら、手順は判断ブロック410に戻る。判断ブロック410への応答がYesならブロック411でアプレットは"removeURL"メッセージを、現在のページのURLをパラメータとしてクライアント・ソフトウェアに送る。ブロック412でアプレットはクライアント・ソフトウェアへの接続を終了する。ブロック413でアプレットはグローバル変数をヌルに戻す。次にアプレットはブロック414で終了する。

【0023】続いて図4を参照する。判断ブロック401への応答がNoならブロック415でアプレットはグローバル変数によって表されるアプレットにURLを追加することを通知する。この時点でアプレットはブロック416、417を通して2つの同時実行パスを開始している。判断ブロック416でアプレットはグローバル変数がヌルかどうか確認する。判断ブロック416への応答がNoなら手順はブロック416に戻る。判断ブロック416への応答がYesなら手順はブロック402に進む。同時にブロック417でアプレットはブラウザから停止メッセージが受信されたかどうか確認する。判断ブロック417への応答がNoなら手順は判断ブロック417に戻る。判断ブロック417への応答がYesならブロック418でアプレットはグローバル変数によって表されるアプレットにこのURLを削除することを通知する。アプレットは次に414で終了する。

【0024】この共同システム内でユーザは、同じページを現在見ているかどうかに応じて互いに共同することができる。クライアント・ソフトウェアは、"addURL"または"removeURL"通知を受け取ったとき、その情報をMUSSSに転送する。各ユーザによって現在表示されているページに関する情報を受け、MUSSSは共同データ（テキスト及びオーディオ・チャット、ホワイトボード・データ等を含む）が、それぞれのブラウザ上で文書と同じように見ているユーザ間で、またはアプリケーションによって決定されるユーザのグループ間で交換されるようにする役割を担う。

【0025】この共同システムを通して、ユーザは"ツアー"に参加することができる。このツアーでは、指定されたリーダが新しいページをリーダのブラウザに"プル"すると、ユーザが見るページに一貫性を持たせるために共同システムが情報を"プッシュ"している間、"フォロー"側のブラウザは同じページを自動的にロードする。このようにしてユーザのグループがWebからコンテンツを同時に"プル"することができる。

【0026】図8を参照する。ユーザがツアーに参加するメカニズムが示してある。ブロック501でユーザは

10

20

30

40

50

11

ツアーのリーダを選択する。ブロック502でクライアント・ソフトウェアはリーダのIDを含む"フォロー"・リクエストをMUSSSに転送する。判断ブロック503でMUSSSはシステムがピアツーピア・モードで動作しているかどうか確認する。判断ブロック503への応答がNoならブロック504でサーバはリーダ及びフォロワの関連付けを格納する。手順は次に505で終了する。判断ブロック503への応答がYesならブロック506でサーバは、ブロック507でフォロワをリーダ自身のフォロワ・リストに追加するリーダに"フォロー"・リクエストを転送する。手順は次に505で終了する。図8には示していないが、ブロック502、503及び506を通るパスは、システムがピアツーピア・モードで動作していることをクライアント・ソフトウェアが確認できる場合は短縮できることは理解されよう。この場合ブロック502-Aに示すように、クライアント・ソフトウェアは"フォロー"・リクエストをリーダに直接転送できる。

【0027】図9を参照する。新しいWebページにツアー・リーダがいつ移動するかがフォロワに通知されるメカニズムが示してある。ブロック601でクライアント・ソフトウェアは"addURL"メッセージをリーダのWebブラウザから受け取る。ブロック602でクライアント・ソフトウェアは"addURL"通知をMUSSSに転送する。判断ブロック603で、システムがピアツーピア・モードで動作しているかどうか確認される。判断ブロック603への応答がNoならブロック604でMUSSSは"changePage"通知をリーダに関連付けられたフォロワのそれぞれに送る。ブロック605でフォロワ側のクライアント・ソフトウェアは"changePage"通知をMUSSSから受信する。ブロック606でクライアント・ソフトウェアは"changePage"通知を、ローカルWebブラウザで実行されているアプレットに転送する。手順は次にブロック607で終了する。判断ブロック603への応答がYesならブロック608でリーダのホスト側のクライアント・ソフトウェアが"changePage"通知をローカル・フォロワ・リストの各フォロワに送る。制御は次に先に述べたようにブロック605に進む。

【0028】従って、共同システムは、全てのクライアントがWebサーバから任意の情報を同時にプルできるようにしながら、全てのクライアントに情報をプッシュすることができる。更に、ブラウザの制御はJavaで書かれたアプレットを通して実行されるので、共同システムは特定のマシンまたはオペレーティング・システム的环境に依存せず、使用中の特定のWebブラウザにも依存しない。

【0029】まとめとして、本発明の構成に関して以下の事項を開示する。

【0030】(1) 情報サーバからデータを取得する方法であって、リクエストが情報サーバからのデータのり

12

クエストを開始するステップと、前記リクエストをプロキシに送るステップと、前記プロキシが前記リクエストを前記情報サーバに転送するステップと、前記プロキシが前記情報サーバからデータを受信し格納するステップと、前記データにアプレットを挿入するステップと、前記データと前記アプレットを前記データの前記リクエストに配信するステップと、前記アプレットを実行するステップと、を含む、方法。

(2) 前記アプレットを実行する前記ステップは、クライアント・ソフトウェアとの接続を開始するステップと、クライアント・ソフトウェアとの前記接続を終了するステップと、を含む、前記(1)記載の方法。

(3) "addURL"通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、前記(2)記載の方法。

(4) "removeURL"通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、前記(2)記載の方法。

(5) "readData"通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、前記(2)記載の方法。

(6) "writeData"通知をクライアント・ソフトウェアとの前記接続を通して送るステップを含む、前記(2)記載の方法。

(7) "changePage"通知をクライアント・ソフトウェアとの前記接続を通して受信するステップと、前記"changePage"通知に従って新しいデータの取得を開始するステップと、を含む、前記(2)記載の方法。

(8) 前記クライアント・ソフトウェアからターゲット・リーダの"フォロー"・リクエストを開始するステップと、前記"フォロー"・リクエストの発信側のIDを記録するステップと、前記ターゲット・リーダによって行われる操作の結果として前記発信側への"changePage"リクエストを開始するステップと、を含む、前記(7)記載の方法。

(9) 前記クライアント・ソフトウェアはマルチユーザ共有状態サーバ(MUSSS)と通信する、前記(2)記載の方法。

(10) クライアント・ソフトウェアとの接続を開始する前記ステップは、"グローバル"・アプレットのIDを確認するステップを含み、前記クライアント・ソフトウェアとの前記接続は、前記"グローバル"・アプレットにリクエストを転送することによって確立される論理接続である、前記(2)記載の方法。

(11) "グローバル"・アプレットの有無を監視するステップと、前記"グローバル"・アプレットが存在しない場合は前記"グローバル"・アプレットになるステップと、を含む、前記(10)記載の方法。

(12) 情報サーバからデータを取得するシステムであって、リクエストが情報サーバからのデータのり

トを開始するための手段と、前記リクエストをプロキシに送る手段と、前記プロキシが前記リクエストを前記情報サーバに転送するための手段と、前記プロキシが前記情報サーバからデータを受信し格納するための手段と、前記データにアプレットを挿入する手段と、前記データと前記アプレットを前記データの前記リクエストに配信する手段と、前記アプレットを実行する手段と、を含む、方法。

(13) 前記アプレットを実行する前記手段は、クライアント・ソフトウェアとの接続を開始する手段と、クライアント・ソフトウェアとの前記接続を終了する手段と、を含む、前記 (12) 記載のシステム。

(14) クライアント・ソフトウェアとの前記接続を通して "addURL" 通知を送る手段を含む、前記 (13) 記載のシステム。

(15) クライアント・ソフトウェアとの前記接続を通して "removeURL" 通知を送る手段を含む、前記 (13) 記載のシステム。

(16) クライアント・ソフトウェアとの前記接続を通して "readData" リクエストを送る手段を含む、前記 (13) 記載のシステム。

(17) クライアント・ソフトウェアとの前記接続を通して "writeData" リクエストを送る手段を含む、前記 (13) 記載のシステム。

(18) クライアント・ソフトウェアとの前記接続を通して "changePage" 通知を受信する手段と、前記 "changePage" 通知に従って新しいデータの取得を開始する手段と、を含む、前記 (13) 記載のシステム。

(19) 前記クライアント・ソフトウェアからターゲット・リーダの "フォロー" リクエストを開始する手段と、前記 "フォロー" リクエストの発信側の ID を記録する手段と、前記ターゲット・リーダによって行われる操作の結果として前記発信側への "changePage" リクエストを開始する手段と、を含む、前記 (18) 記載のシステム。

(20) 前記クライアント・ソフトウェアはマルチユーザ共有状態サーバと通信する、前記 (13) 記載のシステム。

(21) クライアント・ソフトウェアとの接続を開始する前記手段は、"グローバル" アプレットの ID を確認する手段を含み、前記クライアント・ソフトウェアとの前記接続は、前記 "グローバル" アプレットにリクエストを転送することによって確立される論理接続である、前記 (13) 記載のシステム。

(22) "グローバル" アプレットの有無を監視する手段と、前記 "グローバル" アプレットが存在しない場合は前記 "グローバル" アプレットになる手段と、を含む、前記 (21) 記載のシステム。

(23) 情報サーバからデータを取得するためにコンピュータ可読媒体に記録されるコンピュータ・プログラム

・プロダクトであって、リクエストが情報サーバからのデータのリクエストを開始するためのコンピュータ可読手段と、前記リクエストをプロキシに送るコンピュータ可読手段と、前記プロキシが前記リクエストを前記情報サーバに転送するためのコンピュータ可読手段と、前記プロキシが前記情報サーバからデータを受信し格納するためのコンピュータ可読手段と、前記データにアプレットを挿入するコンピュータ可読手段と、前記データ及び前記アプレットを前記データの前記リクエストに配信するコンピュータ可読手段と、前記アプレットを実行するコンピュータ可読手段と、を含む、コンピュータ・プログラム・プロダクト。

(24) 前記アプレットを実行する前記コンピュータ可読手段は、クライアント・ソフトウェアとの接続を開始するコンピュータ可読手段と、クライアント・ソフトウェアとの前記接続を終了するコンピュータ可読手段と、を含む、前記 (23) 記載のコンピュータ・プログラム・プロダクト。

(25) クライアント・ソフトウェアとの前記接続を通して "addURL" 通知を送るコンピュータ可読手段を含む、前記 (24) 記載のコンピュータ・プログラム・プロダクト。

(26) クライアント・ソフトウェアとの前記接続を通して "removeURL" 通知を送るコンピュータ可読手段を含む、前記 (24) 記載のコンピュータ・プログラム・プロダクト。

(27) クライアント・ソフトウェアとの前記接続を通して "readData" リクエストを送るコンピュータ可読手段を含む、前記 (24) 記載のコンピュータ・プログラム・プロダクト。

(28) クライアント・ソフトウェアとの前記接続を通して "writeData" リクエストを送るコンピュータ可読手段を含む、前記 (24) 記載のコンピュータ・プログラム・プロダクト。

(29) クライアント・ソフトウェアとの前記接続を通して "changePage" 通知を受信するコンピュータ可読手段と、前記 "changePage" 通知に従って新しいデータの取得を開始するコンピュータ可読手段と、を含む、前記 (24) 記載のコンピュータ・プログラム・プロダクト。

(30) 前記クライアント・ソフトウェアからターゲット・リーダの "フォロー" リクエストを開始するコンピュータ可読手段と、前記 "フォロー" リクエストの発信側の ID を記録するコンピュータ可読手段と、前記ターゲット・リーダによって行われる操作の結果として前記発信側への "changePage" リクエストを開始するコンピュータ可読手段と、を含む、前記 (29) 記載のコンピュータ・プログラム・プロダクト。

(31) 前記クライアント・ソフトウェアはマルチユーザ共有状態サーバと通信する、前記 (24) 記載のコンピュータ・プログラム・プロダクト。

(32) クライアント・ソフトウェアとの接続を開始する前記コンピュータ可読手段は、"グローバル"・アプレットのIDを確認するコンピュータ可読手段を含み、前記クライアント・ソフトウェアとの前記接続は、前記"グローバル"・アプレットにリクエストを転送することによって確立される論理接続である、前記(24)記載のコンピュータ・プログラム・プロダクト。

(33) "グローバル"・アプレットの有無を監視するコンピュータ可読手段と、前記"グローバル"・アプレットが存在しない場合は前記"グローバル"・アプレットになるコンピュータ可読手段と、を含む、前記(32)記載のコンピュータ・プログラム・プロダクト。

【図面の簡単な説明】

【図1】本発明の方法及びシステムを実現するため利用できるデータ処理システムを示す図である。

【図2】Webブラウザ及びプラットフォームに依存しない形で統合する共同システムのデータ処理要素を示す図である。

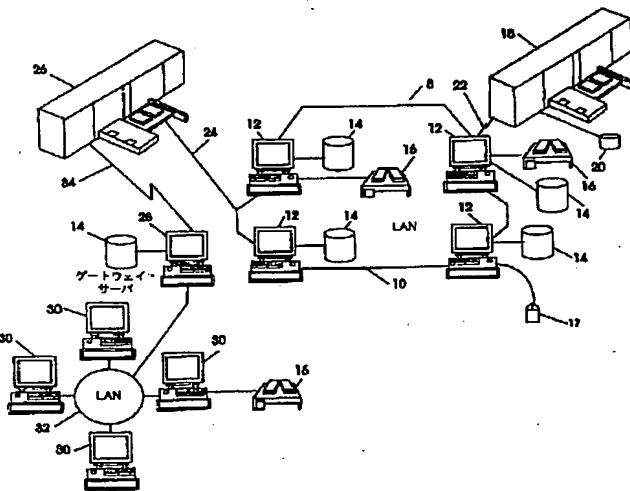
【図3】共同環境でWebサーバからWebブラウザにデータを配信するため必要なステップを示すフローチャートを示す図である。

【図4】Webブラウザで実行されたときに挿入されたアプレットによって実行される操作を示すフローチャートを示す図である。

【図5】Webブラウザで実行されたときに挿入されたアプレットによって実行される操作を示すフローチャートを示す図である。

【図6】Webブラウザで実行されたときに挿入されたアプレットによって実行される操作を示すフローチャートを示す図である。

【図1】



トを示す図である。

【図7】Webブラウザで実行されたときに挿入されたアプレットによって実行される操作を示すフローチャートを示す図である。

【図8】ユーザがツアーにどのように参加するかを示すフローチャートを示す図である。

【図9】ツアー・リーダが新しいWebページに移動するときユーザにどのように通知されるかを示すフローチャートを示す図である。

【符号の説明】

8 データ処理システム

10、32 LAN (ローカル・エリア・ネットワーク)

12、30 個別のコンピュータ

14、20 記憶装置

16 プリンタ/処理装置

17 マウス

18 メインフレーム・コンピュータ

22、24、34 通信リンク

26 サブシステム制御装置/通信コントローラ

28 ゲートウェイ・サーバ

201 MUSSS マルチユーザ共有状態サーバ

202 プロキシ

203 クライアント・ソフトウェア

204 転送ソフトウェア

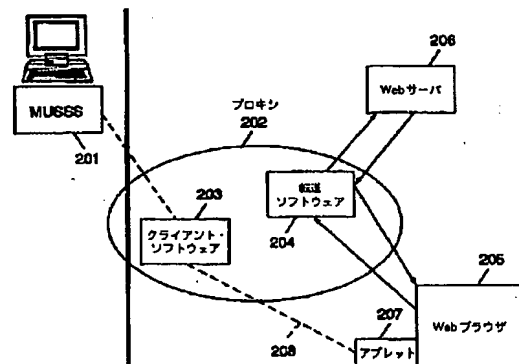
205 ローカルWebブラウザ

206 Webサーバ

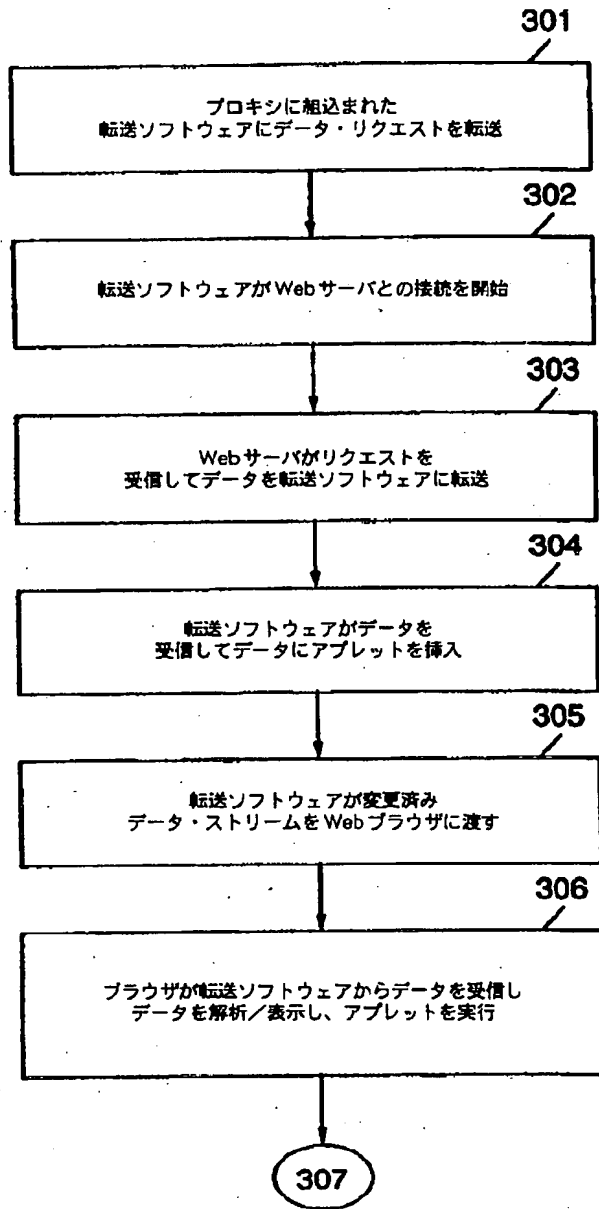
207 アプレット

208 接続

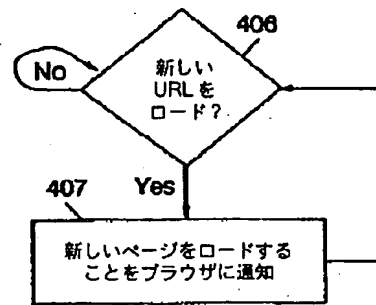
【図2】



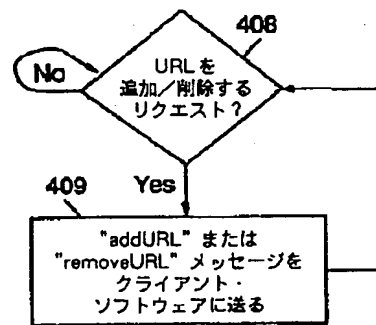
【図3】



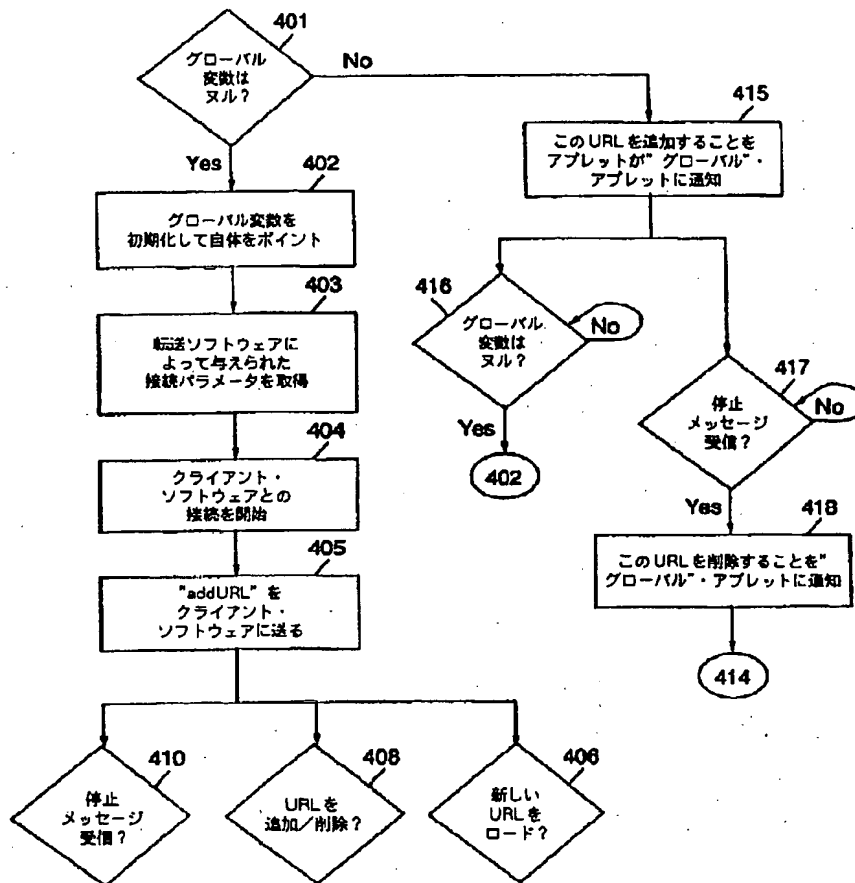
【図5】



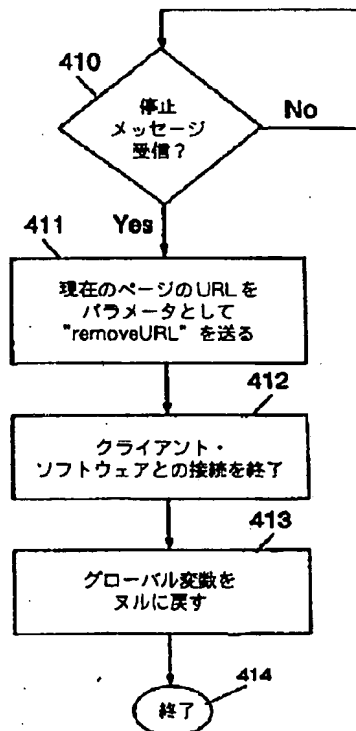
【図6】



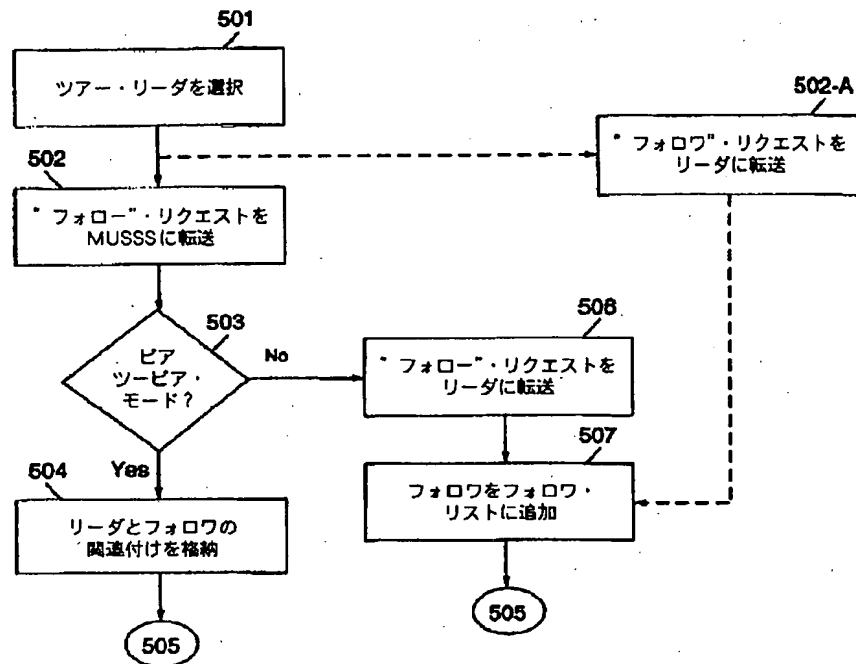
【図4】



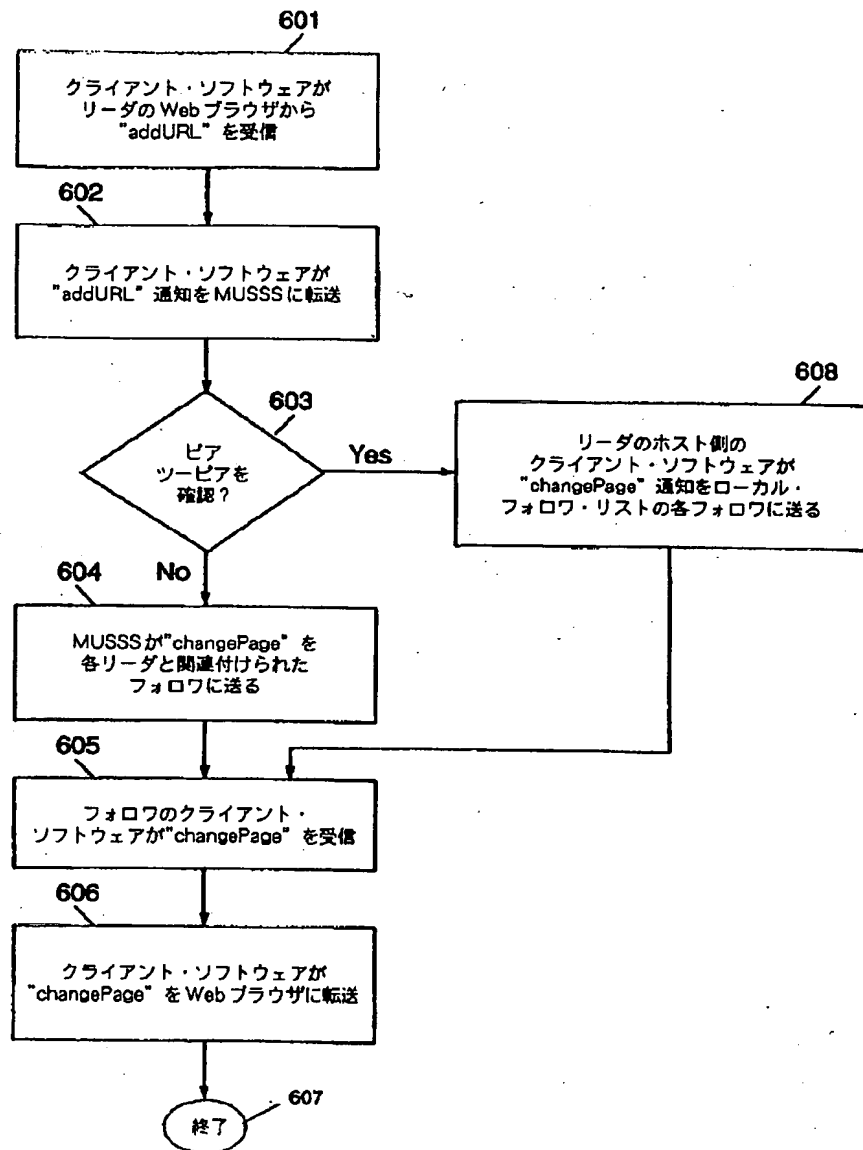
【図7】



【図 8】



【図9】



フロントページの続き

(72)発明者 ビン・キュー・グエン
アメリカ合衆国27513、ノース・カロライ
ナ州カリー、ベントン・ドライブ
305

(72)発明者 ハン・ディ・グエン
アメリカ合衆国27511、ノース・カロライ
ナ州カリー、タローウェイ・ドライブ
2002

(72)発明者 サンディーブ・ケイ・シンハル
アメリカ合衆国27605-1271、ノース・カ
ロライナ州ローリー、ブリッグサースト・
ドライブ、ナンバー 105、1021